

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica ed in Ingegneria dei Sistemi Informatici
Corso di Progettazione del Software
Esame del **22 febbraio 2013**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda una versione modificata del *paintball con doppiogiochisti*. Un gioco è caratterizzato da un nome e da un numero arbitrario, ma superiore a 3, di squadre. Le squadre appartengono esattamente ad un gioco. Ad una squadra partecipa un insieme ordinato, di cardinalità superiore a 2, di giocatori detti *membri regolari* della squadra. Ogni giocatore ha un nome ed è membro regolare di esattamente una squadra. Alcuni giocatori, detti *doppiogiochisti*, sono "membri segreti" di esattamente un'altra squadra. Per ogni giocatore è di interesse ricordare, per ciascuna squadra S_i separatamente, quante volte il giocatore viene colpito dai membri regolari S_i .

Un giocatore è inizialmente *non in gioco*. Quando riceve l'evento *inizio partita* passa allo stato *in gioco*. Ciascun giocatore in gioco, quando riceve un evento *colpo*, verifica a quale squadra appartiene chi ha lanciato il colpo e incrementa il numero di colpi subiti da detta squadra; poi rilancia un evento *colpo* ad un altro giocatore del gioco scelto arbitrariamente ma appartenente ad una squadra diversa da quella di cui è membro regolare. Se il giocatore è un doppiogiochista allora evita di rilanciare l'evento *colpo* anche ai giocatori della squadra di cui è membro segreto¹. Quando un giocatore riceve l'evento *fine partita*, torna nello stato *non in gioco*. Il giocatore può essere modificato solo quando è in *non in gioco*.

Siamo interessati a progettare l'attività del gioco, che prende come parametro un gioco completo delle squadre e dei loro giocatori. Viene prima stampato un messaggio di benvenuto. Quindi, contemporaneamente, si procede a: (i) giocare, attraverso l'invio in broadcasting a tutti i giocatori dell'evento *inizio partita* e dell'ulteriore invio dell'evento *colpo* a 5 giocatori scelti arbitrariamente (tale evento *colpo* ha come mittente i giocatori stessi che lo subiscono), mettendosi poi in attesa (attraverso un'opportuna attività di input/output) del comando di fine esecuzione da parte dell'utente, che termina il gioco riportando tutti i giocatori nello stato "non in gioco"; (ii) si calcola e poi si visualizza su una finestra un report su tutte le squadre, elencando il nome della squadra e i nomi dei suoi membri regolari; (iii) si calcola e visualizza su un'altra finestra separata un report dei giocatori doppiogiochisti, elencando la squadra di cui ciascuno è membro regolare e quella di cui è membro segreto. Una volta terminate queste sottoattività, si verifica quali sono le squadre i cui giocatori hanno ricevuto complessivamente meno colpi e le si eleggono vincitrici, visualizzando i loro nomi.

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare la fase di analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo del diagramma delle classi, diagramma stati e transizioni per la classe *Giocatore*, diagramma delle attività, specifica del diagramma stati e transizioni, e specifica della sola attività principale, motivando, qualora ce ne fosse bisogno, le scelte effettuate.

Domanda 2. Effettuare la fase di progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio definire solo le responsabilità sulle associazioni del diagramma delle classi.

Domanda 3. Effettuare la fase di realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- Le classi *Giocatore* ed eventuali sottoclassi (incluse eventuali classi *fred* corrispondenti);
- Le *associazioni* che legano dette classi alla classe *Squadra*;
- L'*attività principale*, ma non le eventuali sottoattività e attività atomiche.

¹NB: Queste funzioni di scelta possono essere realizzate da funzioni ausiliarie di cui non interessa il corpo, ma solo la segnatura.